Package ‘OvRSeq’

November 23, 2023

Title Ovarian Cancer RNA-Seq Analysis Package

Version 1.0.0

Description

OvRSeq is an R package for analyzing RNA sequencing data from ovarian cancer patients.
The package includes functions for quality control, normalization, differential gene expression analysis,
and pathway analysis. OvaRSeq also provides options for visualization of results, such as heatmaps and volcano plots.
The package is designed to be user-friendly and applicable to various types of RNA sequencing data.

License MIT

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0),

Imports ggplot2,
ggExtra,
SummarizedExperiment,
S4Vectors,
caret,
consensusOV,
GSVA,
immunedeconv,
AnnotationDbi,
biomaRt,
ensembldb,
IRanges,
org.Hs.eg.db,

Suggests knitr,
rmarkdown,

RoxygenNote 7.2.3

R topics documented:

`avg_expression_for_signature_se`

Compute average expression values for gene signatures and enrich colData of a SummarizedExperiment object

---

Description

Given a ‘SummarizedExperiment‘ object and a matrix or data frame of gene signatures, this function computes the average expression values for each gene signature and enriches the ‘colData‘ of the ‘SummarizedExperiment‘ object with the computed values.

Usage

```
avg_expression_for_signature_se(se, gmt)
```

Arguments

`se`	A ‘SummarizedExperiment‘ object with gene expression data.
`gene_sig`	A matrix or data frame where each column represents a gene signature and each cell contains a gene symbol. Empty cells should be represented as empty strings.

Value

A ‘SummarizedExperiment‘ object enriched with new columns in ‘colData‘, each representing the average expression value of a given gene signature for each sample.

Examples

```
# Assuming 'se' is a SummarizedExperiment object and 'gene_sig' is your gene signatures matrix
enriched_se <- avg_expression_for_signature_se(se, gene_sig, "MyGeneSet")
```

`brcaness_classifier` *Random Forest Classifier for BRCAness*

Description

This data object contains a trained random forest classifier that utilizes a 24-gene expression signature to classify patients into two categories: those with BRCAness and those with BRCA status. The classifier was trained using multiomics data, including RNA-Seq data, from the TCGA-OV dataset.

Usage

```
data(brcaness_classifier)
```

Format

A list containing a trained random forest classifier object.

Details

A trained random forest classifier for predicting BRCAness status using a 24-gene expression signature. This classifier was trained with multiomics data from the TCGA-OV dataset and utilizes RNA sequencing (RNA-Seq) data to classify patients as either having BRCAness or BRCA status.

The random forest classifier in this data object was trained with feature selection to optimize its ability to predict BRCAness status based on RNA-Seq data. It is a result of a multiomics approach aimed at accurately classifying patients.

Examples

```
# Load the trained BRCAness classifier
data(brcaness_classifier)

# Predict BRCAness status for a new patient using the classifier
new_patient_data <- ...  # Replace with new patient's RNA-Seq data
prediction <- predict(brcaness_classifier, new_patient_data)
```

`brcaness_signature` *BRCAness Gene Signature*

Description

The BRCAness gene signature is a result of an extensive feature selection process using machine learning models. It represents a subset of genes that are critical for discriminating between BRCAness and non-BRCAness samples based on gene expression data from the TCGA-OV dataset.

Usage

```
data(brcaness_signature)
```

Format

A vector containing the gene symbol for 24 genes.

Details

A gene signature developed through feature selection to optimally classify TCGA-OV samples into BRCAness and non-BRCAness categories. The signature consists of gene expression values (2TPM+1 normalized) for 24 genes. These genes were selected based on their importance in three different machine learning models (Random Forest, Ada Boost, and Gradient Boosting) after recursive feature elimination. Only genes that were among the top 50 most important features in at least two of the three models were included in this signature.

See Also

`TCGA_OV` for the dataset used for feature selection and training.

Examples

```
# Load the BRCAness gene signature
data(brcaness_signature)

# Access gene expression values for the first gene
brcaness_signature
```

calculateIPS *Calculate Immunophenoscore (IPS) from a SummarizedExperiment*

Description

This function calculates the Immunophenoscore (IPS) and its components scores from gene expression data encapsulated in a ‘SummarizedExperiment‘ object. It requires a specific set of genes and corresponding weights provided in a separate file.

Usage

```
calculateIPS(se)
```

Arguments

`se`	A ‘SummarizedExperiment‘ object containing normalized gene expression data.

Value

A data frame with samples as rows and calculated scores (WG, MHC, CP, EC, SC, MDSC, TREG, AZ, IPS) as columns.

References

Immunophenogram: This R-script can be used to calculate Immunophenoscore (IPS) and generate Immunophenogram from "EXPR.txt" and "IPS_genes.txt". The script and associated documentation can be found at the following URL: https://github.com/icbi-lab/Immunophenogram (C) ICBI, Medical University of Innsbruck, Biocenter, Division of Bioinformatics Version 1.0 08.07.2016

classifier_infiltration_status *Tumor Infiltration Status Classifier*

Description

A Random Forest classifier trained on gene expression data from the ICON7 trial. This classifier is designed to classify the tumor immune infiltration status into categories such as ’infiltrated’, ’excluded’, or ’desert’ based on a gene signature. The model is trained using ‘randomForest‘ package and can be used to predict the infiltration status of ovarian cancer samples.

Usage

```
data(classifier_infiltration_status)
```

Format

An object of class ‘randomForest‘ (inherits from ‘list‘), representing a fitted Random Forest model.

Details

The classifier was trained using a Random Forest algorithm with 300 trees on the ICON7 dataset, which contains gene expression data for ovarian cancer samples. The gene signature used for the training consists of genes identified as common between the SummarizedExperiment datasets and the tumor immune phenotype signature.

The ‘classifier_infiltration_status‘ is saved as an R object of class ‘randomForest‘, which contains the entire fitted model object. This model can be used to predict infiltration status in other datasets by supplying the appropriate gene expression data for the genes included in the signature.

The classifier should be applied only to data that has been preprocessed in the same manner as the ICON7 trial data to ensure the validity of the predictions.

References

Desbois M, Udyavar AR, Ryner L, Kozlowski C, Guan Y, Dürrbaum M, et al. Integrated digital pathology and transcriptome analysis identifies molecular mediators of T-cell exclusion in ovarian cancer. Nat Commun. 2020;11:5583.

Examples

```
data(classifier_infiltration_status)
# Suppose `new_data` is a matrix of gene expression values with rows as genes and columns as samples:
predicted_status <- predict(classifier_infiltration_status, new_data)
```

`classify_brcaness` *Classify Samples Using BRCAness Classifier*

Description

This function uses the BRCAness classifier to predict the BRCAness status of the samples in a SummarizedExperiment object. It first checks that all the genes in the BRCAness signature are present in the count data of the SummarizedExperiment object before making predictions.

Usage

```
classify_brcaness(se, brcaness_classifier, brcaness_signature)
```

Arguments

`se` A SummarizedExperiment object with count data.

`brcaness_classifier`
A random forest classifier object trained on BRCAness samples.

`brcaness_signature`
A vector of gene symbols representing the BRCAness gene signature.

Value

A SummarizedExperiment with BRCAness predictions, in ColData.

classify_infiltration_status *Classify Tumor Immune Infiltration Status*

Description

This function applies a trained classifier to a SummarizedExperiment object to classify samples based on their tumor immune infiltration status using a specified gene signature.

Usage

```
classify_infiltration_status(se, classifier, gene_signature)
```

Arguments

`se`	A SummarizedExperiment object containing gene expression data where samples are columns and genes are rows.
`classifier`	A trained classifier object capable of making predictions based on the gene expression data provided in ‘se‘.
`gene_signature`	A character vector containing gene symbols that make up the gene signature used by the classifier. The function will subset the expression data in ‘se‘ based on these genes to make predictions.

Details

The function first validates that the provided ‘se‘ object is indeed a SummarizedExperiment. It then extracts the expression data for the genes in the ‘gene_signature‘ from the ‘se‘ object. This gene expression matrix is transposed (samples as rows, genes as columns) and used as input for the classifier to predict the infiltration status. The predictions are then added to the ‘colData‘ of the ‘se‘ object, allowing for easy retrieval and further analysis.

Value

The SummarizedExperiment object ‘se‘ with an additional column in ‘colData‘ named ‘InfiltrationStatus‘, which contains the classification results for each sample.

Examples

```
# Assuming `se` is a SummarizedExperiment object with gene expression data,
# `rf_classifier` is a trained random forest classifier, and
# `immune_genes` is a vector of gene symbols making up the immune signature:
se <- classify_infiltration_status(se, rf_classifier, immune_genes)
```

computeCytC1qaRatio *Compute CYT to C1QA Ratio (C2C)*

Description

This function calculates the cytolytic activity (CYT) to C1QA ratio (C2C) for a given SummarizedExperiment object. The ratio is calculated using the expression values of GZMB, PRF1, and C1QA. The C2C ratio is then added to the ‘colData‘ of the provided SummarizedExperiment object.

Usage

```
computeCytC1qaRatio(se)
```

Arguments

se	A SummarizedExperiment object containing gene expression data. The object must contain expression data for GZMB, PRF1, and C1QA genes.

Details

The function computes the C2C ratio using the formula: C2C = 0.5 x (log2(TPM + 1 of GZMB) + log2(TPM + 1 of PRF1)) / log2(TPM + 1 of C1QA). It assumes that the expression data in the ‘SummarizedExperiment‘ object are either TPM (Transcripts Per Million) or intensity values suitable for the log transformation.

Value

Returns the SummarizedExperiment object with an additional column in ‘colData‘ named ‘ratio_CYT_C1QA‘, representing the computed C2C ratio.

Examples

```
# se is a SummarizedExperiment object with expression data for GZMB, PRF1, and C1QA
updatedSe <- computeCytC1qaRatio(se)
```

deconvolute_immune *Deconvolute Immune Cell Fractions from Gene Expression Data*

Description

This function deconvolutes immune cell fractions from gene expression data using methods from the ‘immunedeconv‘ package.

Usage

```
deconvolute_immune(se, method)
```

Arguments

`se` A Summarized Experiment of gene expression values (logTPM+1), where rows are genes and columns are samples.

`method` A character string indicating the deconvolution method to be used. See details for available methods.

Details

Available methods and their respective licensing and citations are as follows:

method	license	citation
quanTIseq	free (BSD)	Finotello et al. (2019) Genome Medicine
TIMER	free (GPL 2.0)	Li et al. (2016) Genome Biology
CIBERSORT	free for non-commerical use only	Newman et al. (2015) Nature Methods
MCPCounter	free (GPL 3.0)	Becht et al. (2016) Genome Biology
xCell	free (GPL 3.0)	Aran et al. (2017) Genome Biology
EPIC	free for non-commercial use only	Racle et al. (2017) ELife

Note: While 'immunedeconv' itself is free (BSD), you may need to obtain a license to use individual methods. Please ensure you cite both this package and the method(s) you use in your work.

Value

A list with deconvolution results.

References

Sturm, G., et al. (2019) Bioinformatics. https://doi.org/10.1093/bioinformatics/btz363

Examples

```
# data <- ... # Your gene expression data (logTPM+1)
# res <- deconvolute_immune(data, method="timer")
```

`generateReport` *Generate OvRSeq Analysis Reports for Patients*

Description

This function generates individual PDF reports for each patient in a given dataset. Each report includes the analysis results for the patient using the OvRSeq pipeline, alongside TCGA (The Cancer Genome Atlas) reference statistics.

Usage

```
generateReport(se, outputDir)
```

Arguments

`se` An object containing the patient sample data.

`outputDir` A string specifying the directory where the PDF reports will be saved.

Value

This function does not return a value, but generates PDF reports and saves them in the specified output directory.

Examples

```
# Assuming 'se' is your sample
# Also assuming you have a valid output directory path in 'outputPath'
generateReport(se = se, tcgaData = tcgaData, outputDir = outputPath)
```

---

get_consensus_ov_subtypes *Obtain Consensus Subtypes of High-Grade Serous Ovarian Cancer*

Description

This function gets the consensus molecular subtypes of high-grade serous (HGS) ovarian cancer using the ‘consensusOV‘ package. The function implements a consensus classifier which consolidates and improves on the robustness of proposed subtype classifiers.

Usage

```
get_consensus_ov_subtypes(
  se,
  ids_type = "symbol",
  concordant.tumors.only = TRUE,
  remove.using.cutoff = FALSE,
  percentage.dataset.removed = 0.75
)
```

Arguments

`se` A Summarized Experiments of gene expression values, with gene symbol or entrez id as identifiers.

`ids_type` A character string indicating the type of IDs used in the ’data’ row names. Can be either "symbol" or "entrez".

`concordant.tumors.only`
Logical. If TRUE, only tumors that are concordantly classified across all datasets are included in the final classification. Defaults to TRUE.

`remove.using.cutoff`
: Logical. If TRUE, tumors with poor classification confidence are removed using the optimal cutoff. Defaults to FALSE.

`percentage.dataset.removed`
: Numeric value indicating the percentage of the dataset to be removed based on classification confidence. Defaults to 0.75.

Value

Summarized Experiments with Tumor_Molecular_Subtypes of consensus molecular subtypes for the samples, in colData.

References

Chen G, Kannan L, Geistlinger L, Kofia V, Safikhani Z, Gendoo D, Parmigiani G, Birrer M, Haibe-Kains B, Waldron L (2018). "Consensus on molecular subtypes of high-grade serous ovarian carcinoma." Clinical Cancer Research, 24, 4990. doi:10.1158/1078-0432.CCR-18-0784.

Examples

```
load_TCGA_OV
TCGA_OV <- get_consensus_ov_subtypes(TCGA_OV, ids_type = "symbol")
```

`immune_signatures` *Immune Signatures*

Description

The ‘immune_signatures‘ object contains well-defined immune-related gene signatures in GMT format. These signatures are valuable for characterizing immune processes and activities in biological data, including gene expression profiles.

Usage

```
data(immune_signatures)
```

Format

A list of lists, where each sublist contains: - ‘Name‘: The name of the immune signature. - ‘Genes‘: A character vector of gene symbols representing the signature genes.

Details

A GMT (Gene Matrix Transposed) file containing well-defined immune-related gene signatures. Each signature consists of a list with the name of the signature and a vector of gene symbols representing the genes that constitute the signature. These signatures are used to characterize immune-related processes, including T cell inflammation, IFN gamma signature, cytolytic activity, and cytotoxic T lymphocyte function.

See Also

For more information on GMT files and gene set enrichment analysis (GSEA), refer to the relevant literature and software documentation.

Examples

```
# Load the immune signatures
data(immune_signatures)

# Access the genes in the "T cell inflammation" signature
t_cell_inflammation_genes <- immune_signatures$T_cell_inflammation$Genes
```

---

immune_signature_score *Calculate enrichment scores for immune signatures*

Description

This function calculates enrichment scores for a list of immune signatures using the GSVA method. The user can provide their own gene list or use one of the pre-defined ones. The function returns an updated SummarizedExperiment object with the enrichment scores added to the colData.

Usage

```
immune_signature_score(se, method = "ssgsea", genelist)
```

Arguments

se — A SummarizedExperiment object containing RNA sequencing data

method — A character string indicating the method to use for calculating enrichment scores. Default is "ssgsea".

genelist — A character vector with the gene list for the immune signature.

Value

A SummarizedExperiment object with the enrichment scores added to the colData.

IPS_genes *Immunophenoscore (IPS) Genes and Weights*

Description

A dataset containing genes and corresponding weights used to calculate the Immunophenoscore (IPS), which is a measure of the immune landscape of a tumor.

Usage

```
data(IPS_genes)
```

Format

A data frame with the following columns:

GENE (character) Official gene symbol.

WEIGHT (numeric) Weight of the gene in the IPS calculation.

CATEGORY (character) The immunological category to which the gene belongs (e.g., MHC, CP, EC, SC, MDSC, TREG, AZ).

Source

Immunophenogram project: https://github.com/icbi-lab/Immunophenogram

References

Charoentong P, Finotello F, Angelova M, Mayer C, Efremova M, Rieder D, Hackl H, Trajanoski Z. Pan-cancer Immunogenomic Analyses Reveal Genotype-Immunophenotype Relationships and Predictors of Response to Checkpoint Blockade. Cell Rep. 2017 Jan 3;18(1):248-262. doi: 10.1016/j.celrep.2016.12.019. PMID: 28052254.

Examples

```
# Load the IPS genes and weights
data(IPS_genes)

# Access the gene symbols and weights
ips_genes <- IPS_genes$GENE
ips_weights <- IPS_genes$WEIGHT
```

load_brcaness_classifier

Load BRCAness Classifier

Description

Loads the pre-trained BRCAness classifier from the "brcaness_classifier.rda" data file and returns it as the ‘brcaness_classifier‘ object.

Usage

```
load_brcaness_classifier()
```

Value

The pre-trained BRCAness classifier as a list.

Examples

```
# Load the BRCAness classifier
brcaness_classifier <- load_brcaness_classifier()
```

load_brcaness_signature

Load BRCAness Gene Signature

Description

Loads the BRCAness gene signature from the "brcaness_signature.rda" data file and returns it as a vector with gene symbol for the for 24 genes.

Usage

```
load_brcaness_signature()
```

Value

A vector with gene symbol for the BRCAness gene signature.

Examples

```
# Load the BRCAness gene signature
brcaness_signature <- load_brcaness_signature()

# Access gene expression values for the first gene
brcaness_signature$Gene1
```

`load_classifier_infiltration_status` *Load Infiltration Status Classifier*

Description

Loads the pre-trained classifier for tumor immune infiltration status from the "classifier_infiltration_status.rda" data file. This classifier is used to determine the infiltration status (e.g., infiltrated, excluded, desert) within the tumor microenvironment.

Usage

```
classifier_infiltration_status <- load_classifier_infiltration_status()
```

Value

A pre-trained classifier object. Typically, this is a machine learning model object that can be used to predict infiltration status based on gene expression profiles.

Examples

```
# Load the Infiltration Status classifier
classifier_infiltration_status <- load_classifier_infiltration_status()

# Now you can use `classifier_infiltration_status` with gene expression data to
# predict infiltration statuses.
```

`load_immune_signatures` *Load Immune Signatures*

Description

Loads the immune signatures from the "immune_signatures.rda" data file and returns them as a list of lists, where each sublist contains the name of an immune signature and a vector of gene symbols representing the genes that constitute the signature.

Usage

```
load_immune_signatures()
```

Value

A list of lists, where each sublist contains: - ‘Name‘: The name of the immune signature. - ‘Genes‘: A character vector of gene symbols representing the signature genes.

Examples

```
# Load the immune signatures
immune_signatures <- load_immune_signatures()

# Access the genes in the "T cell inflammation" signature
t_cell_inflammation_genes <- immune_signatures$T_cell_inflammation$Genes
```

load_small_immune_signatures *Load Small Immune Signatures*

Description

Loads the small immune signatures from the "small_immune_signatures.rda" data file and returns them as a list of lists. Each sublist corresponds to an immune signature that contains fewer than 10 genes, making it suitable for analysis methods like z-score computations that may be more appropriate for smaller gene sets than single-sample gene set variance analysis.

Usage

```
small_immune_signatures <- load_small_immune_signatures()
```

Value

A list of lists, where each sublist contains: - ‘Name‘: The name of the small immune signature. - ‘Genes‘: A character vector of gene symbols representing the genes in the signature.

Examples

```
# Load the small immune signatures
small_immune_signatures <- load_small_immune_signatures()

# Access the genes in the "T cell inflammation" signature
ifng_ayers <- small_immune_signatures$'IFNG Ayers'
```

load_TCGA_OV *Load TCGA-OV Dataset*

Description

Loads the TCGA-OV dataset from the "TCGA_OV.rda" file and returns it as a SummarizedExperiment object.

Usage

```
load_TCGA_OV()
```

Value

A SummarizedExperiment object containing raw RNA-Seq counts from the TCGA-OV dataset with associated metadata columns.

Examples

```
# Load the TCGA-OV dataset
my_dataset <- load_TCGA_OV()

# Access metadata columns
metadata(my_dataset)$AGE
metadata(my_dataset)$TUMOR_GRADE
```

load_tumor_immune_phenotype_signature
Load Tumor Immune Phenotype Signature

Description

Loads the tumor immune phenotype signature from the "tumor_immune_phenotype_signature.rda" data file. This signature contains a list of genes associated with the immune phenotype of tumors.

Usage

```
tumor_immune_phenotype_signature <- load_tumor_immune_phenotype_signature()
```

Value

A character vector of gene symbols representing the tumor immune phenotype signature.

Examples

```
# Load the tumor immune phenotype signature
tumor_immune_phenotype_signature <- load_tumor_immune_phenotype_signature()
```

OvaRSeqDataSet *OvaRSeq constructor*

Description

OvaRSeq constructor

Usage

```
OvaRSeqDataSet(countData, colData)
```

Arguments

countData	a matrix or data frame contains gene count
colData	a DataFrame or data.frame
...	optional arguments passed to SummarizedExperiment

Value

a OvaRSeq object

OvaRSeqDataSet-class *OvaRSeqDataSet class*

Description

OvaRSeqDataSet is a class inherited from SummarizedExperiment. It is used to store the count matrix, colData, and design formula in differential expression analysis.

References

Martin Morgan, Valerie Obenchain, Jim Hester and Hervé Pagès (2018). SummarizedExperiment: SummarizedExperiment container. R package version 1.12.0.

OvRSeq	*Comprehensive RNA Sequencing-based Characterization of HGSOC Samples*

Description

This function provides an all-encompassing analysis of high-grade serous ovarian cancer (HGSOC) RNA-seq data. It integrates various predictive biomarkers and performs a multi-faceted immune profiling, leveraging a 24-gene expression signature to classify BRCAness, subtype classification, immune environment profiling, and immune cell deconvolution.

Usage

```
OvRSeq(se, normalize = FALSE)
```

Arguments

`se`	A SummarizedExperiment object containing RNA-seq data for HGSOC samples.
`normalize`	A logical parameter indicating whether the assay data should be normalized. If FALSE, the function will expect raw count data in integer form.

Value

Returns a SummarizedExperiment object enriched with BRCAness classification, molecular subtype classification, immune phenotyping, IPS scoring, immune signatures, and immune cell deconvolution estimates.

Examples

```
# Load data
se <- load_TCGA_OV()
# Run the OvRSeq analysis pipeline
se <- OvRSeq(se)
```

plot_ggmarginal	*Plot Marginal Distributions for OvRSeq Results*

Description

This function takes the results from OvRSeq, extracts specified variables from the column data, and creates a ggplot with marginal histograms to show the distribution of x and y variables separately.

Usage

```
plot_ggmarginal(se, x_var, y_var, color_var = NA)
```

Arguments

`se` A SummarizedExperiment object containing results from OvRSeq.

`x_var` The name of the variable in colData(se) to use for the x-axis.

`y_var` The name of the variable in colData(se) to use for the y-axis.

`color_var` The name of the variable in colData(se) to use for coloring the points.

Value

A ggplot object with added marginal histograms.

Examples

```
# Assuming `se` is a SummarizedExperiment object with relevant data
# plot_ggmarginal(se, "variable1", "variable2", "groupingVar")
```

`plot_vulnerabilitymap` *Plot Vulnerability Map*

Description

This function creates a vulnerability map based on three key variables: the vulnerability score, the BRCAness score, and the cytolytic activity (CYT) to C1QA ratio (C2C). The BRCAness score is derived from the prediction probability of a random forest classifier. The C2C ratio is calculated using the log2-transformed expression values of GZMB, PRF1, and C1QA.

Usage

```
plot_vulnerabilitymap(se)
```

Arguments

`se` A SummarizedExperiment object containing the necessary data for plotting. It must include BRCAness probabilities and CYT to C1QA ratios, or data to compute them.

Details

The vulnerability score is defined as a weighted sum of the BRCAness probability and the C2C ratio, where the weights are determined by a logistic regression model. This model uses log2 intensity expression values and SigMA status (mutational signature 3) data from the TOPACIO cohort with the treatment response as a binary dependent variable.

The function plots a two-dimensional map with the transformed C2C ratio and BRCAness probability as coordinates. The map is color-coded based on the vulnerability score.

C2C is computed as follows:

$$C2C = 0.5 \times (\log_2(GZMB + 1) + \log_2(PRF1 + 1)) / \log_2(C1QA + 1)$$

The vulnerability score is computed using:

$$VulnerabilityScore = 2.687 \times BRCAnessProbability + 1.051 \times C2C$$

For visualization, C2C values are transformed to a range between 0 and 1 using a sigmoid function. The BRCAness probability and the transformed C2C values are used as coordinates on the map.

Value

A ggplot object representing the vulnerability map.

Examples

```
# se is a SummarizedExperiment object with required data
plot_vulnerabilitymap(se)
```

---

small_immune_signatures

Small Immune Signatures

Description

The ‘small_immune_signatures‘ object contains well-curated, small immune-related gene signatures in GMT format, suitable for refined gene expression analysis. These small signatures are derived from larger gene sets, trimmed for optimal focus and precision in immune process characterization.

Usage

```
data(small_immune_signatures)
```

Format

A list of lists, where each sublist contains: - ‘Name‘: The name of the immune signature. - ‘Genes‘: A character vector of gene symbols, each representing fewer than 10 signature genes for focused analysis.

Details

A GMT (Gene Matrix Transposed) file containing concise immune-related gene signatures. Each signature is limited to fewer than 10 genes, which is why they are termed "small." These limited gene sets can provide more precise z-score calculations for immune-related processes as opposed to single-sample gene set enrichment variances.

Each signature within the file facilitates the characterization of specific immune-related activities, such as T cell inflammation, interferon-gamma response, cytolytic activity, cytotoxic T lymphocyte response, and various components of the immune system. Splitting larger signatures into smaller, more manageable sets can yield a clearer understanding of the biological data by focusing on the most relevant and influential genes.

See Also

For more detailed information on the use of small gene sets for z-score calculations in immune signature analysis, as well as general GMT file structure and gene set enrichment analysis (GSEA), refer to the relevant literature and software documentation.

Examples

```
# Load the small immune signatures
data(small_immune_signatures)

# Access the genes in the "T cell inflammation" signature
t_cell_inflammation_genes <- small_immune_signatures$T_cell_inflammation$Genes
```

---

ssGSEA_OV_custom *Perform Single-Sample Gene Set Enrichment Analysis (ssGSEA) on a SummarizedExperiment*

Description

This function performs ssGSEA on a given ‘SummarizedExperiment‘ object using custom gene sets provided in GMT format. It leverages the ‘GSVA‘ package to compute enrichment scores for each sample in the experiment. GSVA is a non-parametric, unsupervised method for estimating variation of gene set enrichment through the samples of an expression dataset.

Usage

```
ssGSEA_OV_custom(se, gmt_list)
```

Arguments

`se`	A ‘SummarizedExperiment‘ object containing the expression data.
`gmt_list`	A list where each element is a character vector of gene symbols representing a gene set. The names of the list elements are used as gene set names.

Value

A matrix of enrichment scores with gene sets as rows and samples as columns.

References

Hänzelmann S, Castelo R, Guinney J (2013). “GSVA: gene set variation analysis for microarray and RNA-Seq data.” BMC Bioinformatics, 14, 7. <doi:10.1186/1471-2105-14-7> For a full list of citations, use ’citation("GSVA")’ in R.

Examples

```
# Assuming 'se' is a SummarizedExperiment object and 'gmt_list' is your list of gene sets
enrichment_scores <- ssGSEA_OV_custom(se, gmt_list)
```

`TCGA_OV` *TCGA-OV RNA-Seq Dataset*

Description

This dataset contains raw RNA-Seq counts from the TCGA-OV dataset. It includes metadata columns that provide additional information about the samples, including patient age, tumor grade, clinical stage, tumor residual disease status, and BRCAness classification.

Usage

```
data(TCGA_OV)
```

Format

A SummarizedExperiment object.

Details

A SummarizedExperiment object containing raw RNA-Seq counts from the TCGA-OV dataset, along with metadata columns including AGE, TUMOR_GRADE, CLINICAL_STAGE, TUMOR_RESIDUAL_DISEASE, and BRCAness.

The TCGA-OV dataset is stored as a SummarizedExperiment object, which provides an organized and efficient structure for working with high-throughput genomics data. The metadata columns allow for the exploration of clinical and molecular attributes associated with the samples.

See Also

`SummarizedExperiment` for more information on SummarizedExperiment objects.

Examples

```
# Load the TCGA-OV dataset
data(TCGA_OV)

# Access metadata columns
metadata(TCGA_OV)$AGE
metadata(TCGA_OV)$TUMOR_GRADE
```

train_rf *Train a random forest model on gene expression data*

Description

This function trains a random forest model on a summarized experiment object containing gene expression data, using a specified gene signature to select a subset of genes, and a specified label in the colData to use as the target variable. The function returns the best random forest model.

Usage

```
train_rf(se, label, gene_signature)
```

Arguments

`se` A SummarizedExperiment object containing gene expression data

`label` A string indicating the name of the column in colData containing the label to use as the target variable for classification

`gene_signature` A character vector containing the names of genes to use in the analysis

Value

A trained random forest model

Examples

```
# Load the TCGA_OV.rda and BRCAness_signature.rda files
data(TCGA_OV)
data("brcaness_signature")

# Train a random forest model using the BRCAness label
rf_model <- train_rf(se, "BRCAness", brcaness_signature)

# Print the model
print(rf_model)

# Make predictions using the model
predictions <- predict(rf_model, newdata = t(assay(TCGA_OV)[rownames(BRCAness_signature),]))

# Print the predictions
print(predictions)
```

`train_rf_classifier_infiltration_status` *Train Random Forest Classifier for Tumor Infiltration Status*

Description

Trains a random forest classifier using a predefined gene signature to predict tumor immune phenotypes, such as infiltrate, excluded, or desert based on the gene expression profiles from a SummarizedExperiment object.

Usage

```
train_rf_classifier_infiltration_status(
  se,
  gene_signature,
  num_estimators = 300
)
```

Arguments

`se`	A SummarizedExperiment object containing gene expression data and a column 'TinfStatus' in its 'colData' which indicates the tumor immune phenotype.
`gene_signature`	A character vector containing the names of genes in the signature to be used for model training.
`num_estimators`	The number of trees to grow in the random forest model. Default is 300.

Details

The function extracts the expression data for the genes in the provided signature from a SummarizedExperiment object and uses the 'TinfStatus' from the column metadata to train a random forest classifier. The 'TinfStatus' should be a factor that represents the tumor immune phenotype status with levels such as 'infiltrate', 'excluded', or 'desert'. The gene expression matrix is transposed to ensure proper dimensions for the random forest training function. The 'train' function from the 'caret' package is used for training the model, with the number of trees set by the 'num_estimators' parameter.

The function sets the seed for reproducibility and ensures the output is a trained model that can predict the 'TinfStatus' based on gene expression profiles.

Value

A random forest model object trained on the gene expression data and tumor immune phenotype status.

References

Desbois M, Udyavar AR, Ryner L, Kozlowski C, Guan Y, Dürrbaum M, et al. Integrated digital pathology and transcriptome analysis identifies molecular mediators of T-cell exclusion in ovarian cancer. Nat Commun. 2020;11:5583.

Examples

```
# Assuming 'se' is a preloaded SummarizedExperiment object with the required data
# and 'tumor_immune_phenotype_signature' is the predefined gene signature list:
rf_classifier <- train_rf_classifier_infiltration_status(se, tumor_immune_phenotype_signature)
```

`tumor_immune_phenotype_signature`

Tumor Immune Phenotype Signature

Description

The ‘tumor_immune_phenotype_signature‘ object contains a gene signature that classifies ovarian cancer tumor immune phenotypes. It was developed using integrated digital pathology and transcriptome analysis, with a focus on CD8+ T cell presence and position within the tumor.

Usage

```
data(tumor_immune_phenotype_signature)
```

Format

A list containing gene symbols representing the 148 genes in the tumor immune phenotype signature.

Details

A gene signature developed based on digital pathology and transcriptome analysis to classify tumor immune phenotypes (infiltrate, excluded, desert) in ovarian cancer. This signature was derived from a classification model using 157 genes that describe the presence and position of CD8+ T cells relative to the tumor center or margin in the ICON7 cohort.

References

Desbois M, Udyavar AR, Ryner L, Kozlowski C, Guan Y, Dürrbaum M, et al. Integrated digital pathology and transcriptome analysis identifies molecular mediators of T-cell exclusion in ovarian cancer. Nat Commun. 2020;11:5583.

Examples

```
# Load the tumor immune phenotype signature
data(tumor_immune_phenotype_signature)

# Access the gene symbols in the signature
signature_genes <- tumor_immune_phenotype_signature
```

`update_se_with_entrez_ids` *Update Gene Symbols to Entrez IDs in a SummarizedExperiment*

Description

This function takes a SummarizedExperiment object with gene symbols and updates the row names with corresponding Entrez IDs.

Usage

```
update_se_with_entrez_ids(se)
```

Arguments

`se`	A SummarizedExperiment object.

Value

A SummarizedExperiment object with Entrez IDs as row names.

Examples

```
# Assuming you have a SummarizedExperiment object named 'se'
se_updated <- update_se_with_entrez_ids(se)
rownames(se_updated)
```

Index